In the Claims

This listing of claims will replace all prior versions and listings of claims in the application:

1. (Previously Amended) A digital system comprising a microprocessor having an instruction execution pipeline with a plurality of pipeline phases, wherein the microprocessor comprises:

program fetch circuitry operable to perform a first portion of the plurality of pipeline phases;

instruction decode circuitry connected to receive fetched instructions from the program fetch circuitry, the instruction decode circuitry operable to perform a second portion of the plurality of pipeline phases; and

at least a first functional unit connected to receive a plurality of control signals from the instruction decode circuitry, the functional unit operable to perform a third portion of the plurality of pipeline phases, the third portion being execution phases, wherein the first functional unit comprises:

first test circuitry connected to receive an operand from a selected test register, and having an output for indicating a condition of the operand;

decrement circuitry connected to receive the operand from the selected test register, and having an output connected to conditionally provide a decremented value of the operand to the test register dependent upon said indicated condition of the operand;

adder circuitry connected to receive a program counter value and a displacement value, and having an output connected to conditionally provide a branch address to a program counter register dependent upon said indicated condition of the operand; and

wherein the first test circuitry, the decrement circuitry, and the adder circuitry are all operable to test the operand, conditionally decrement the operand, and conditionally provide a branch address to the program counter in response to a single conditional branch-decrement instruction.

2. (Previously Amended) The digital system of Claim 1, wherein the first test circuitry, the decrement circuitry, and the adder circuitry are all operable to test the operand, conditionally decrement the operand, and conditionally provide a branch address to the program counter in response to a single conditional branch-decrement instruction during a single one of the third portion of pipeline phases.

3. (Canceled)

4. (Currently Amended) The digital system of Claim ~~3~~ 1, further comprising second test circuitry connected to test a condition of a selected predicate register, and having an output for indicating a condition of the predicate register, wherein the second test circuitry is operable to inhibit the program counter from receiving the branch address if the contents of the predicate register do not correspond to a second condition.

Claims 5 to 9. (Canceled)

10. (Currently Amended) A method of operating a digital system having a microprocessor with a conditional branch-decrement instruction, comprising the steps of:

fetching a conditional branch-decrement instruction for execution;

testing a test register selected by the conditional branch-decrement instruction from among a plurality of distinct data registers to determine if the contents of the test register meet a first condition;

providing a branch address to a program counter to cause a branch if the contents of the test register meet the first condition; and

modifying the contents of the test register if the contents of the test register meet the first condition.

11. (Previously Amended) The method of Claim 10, further comprising the steps of:

testing a predicate register selected by the conditional branch-decrement instruction to determine if the contents of the predicate register meet a second condition; and

inhibiting the step of providing a branch address to the program counter and inhibiting said step of modifying the contents of the test register if the contents of the predicate register do not meet the second condition.

12. (Canceled)

13. (Original) The method of Claim 10, wherein the steps of testing, providing, and modifying are all performed during a same execution phase of the microprocessor.

14. (Canceled)

15. (Previously Added) The digital system of Claim 1, further comprising:

a register file including a plurality of general purpose registers, each general purpose register capable of supplying an

operand to a functional unit and capable of receiving destination data generated by a functional unit; and

wherein said conditional branch-decrement instruction designates one of said general purpose registers as said selected test register.

16. (Previously Added) The digital system of Claim 15, further comprising:

second test circuitry connected to test a condition of a selected predicate register, and having an output for indicating a condition of the predicate register, wherein the second test circuitry is operable to inhibit the program counter from receiving the branch address and inhibit said step of modifying the contents of the test register if the contents of the predicate register do not correspond to a second condition; and

wherein said conditional branch-decrement instruction designates one of said general purpose registers as said predicate register.

17. (Previously Added) The digital system of Claim 16, wherein:

said conditional branch-decrement instruction designates one of said general purpose registers of a predetermined subset of said general purpose registers as said predicate register.

18. (Previously Added) The digital system of Claim 1, wherein:

said program fetch circuitry operable to fetch a fetch packet of a predetermined plurality of instructions each first portion of the plurality of pipeline phases starting at predetermined address boundaries; and

said adder circuitry adds said displacement value to a last predetermined address boundary.

19. (Previously Added) The digital system of claim 18, wherein:

said instruction decode circuitry reads a predetermined bit of each instruction to determine an execute packet of instructions capable of execution in parallel on a plurality of functional units, wherein an execute packet may include instructions in two sequential fetch packets; and

said adder circuitry adds said displacement value to said last predetermined address boundary of said fetch packet of said conditional branch-decrement instruction.

20. (Currently Amended) The method of Claim 10, further comprising the ~~step~~ steps of:

storing data in a register file including a plurality of distinct general purpose registers;

recalling data from an instruction designated general purpose register for supplying an operand to a functional unit;

storing destination data generated by a functional unit in an instruction designated general purpose register; and

designating via the conditional branch-decrement instruction one of said general purpose registers as said selected test register.

21. (Previously Added) The method of Claim 20, further comprising:

testing a predicate register selected by the conditional branch-decrement instruction to determine if the contents of the predicate register meet a second condition; and

designating via the conditional branch-decrement instruction one of said general purpose registers as said predicate register.

22. (Previously Added) The method of Claim 21, wherein:

said step of designating said predicate register designates said predicate register from a predetermined subset of said general purpose registers as said predicate register.

23. (Previously Added) The method of Claim 10, wherein:

said step of fetching instructions fetches a fetch packet of a predetermined plurality of instructions; and

said step of providing a branch address to the program counter adds a displacement value to a last predetermined address boundary.

24. (Previously Added) The method of claim 23, wherein:

reading a predetermined bit of each instruction to determine an execute packet of instructions capable of execution in parallel on a plurality of functional units, wherein an execute packet may include instructions in two sequential fetch packets;

dispatching each instruction of each execute packet to a corresponding functional unit in parallel;

said step of providing a branch address to the program counter adds said displacement value to a last predetermined address boundary of a second sequential fetch packet if said second sequential fetch packet contains said conditional branch-decrement instruction.

25. (New) A method of operating a digital system having a microprocessor with a conditional branch-decrement instruction, comprising the steps of:

fetching a conditional branch-decrement instruction for execution;

testing a test register selected by an operand field of the conditional branch-decrement instruction to determine if the contents of the test register meet a first condition;

providing a branch address to a program counter to cause a branch if the contents of the test register meet the first condition; and

modifying the contents of the test register if the contents of the test register meet the first condition.

26. (New) The method of Claim 25, further comprising the steps of:

storing data in a register file including a plurality of distinct general purpose registers;

recalling data from an instruction designated general purpose register for supplying an operand to a functional unit;

storing destination data generated by a functional unit in an instruction designated general purpose register; and

designating via the conditional branch-decrement instruction one of said general purpose registers as said selected test register.

27. (New) The method of Claim 25, wherein:

said step of providing a branch address to the program counter to cause a branch combines a displacement field of the conditional branch-decrement instruction with current contents of the program counter.

28. (New) The method of Claim 25, wherein:

said step of providing a branch address to the program counter adds a signed displacement designated by the displacement field of the conditional branch-decrement instruction to current contents of the program counter.

29. (New) The method of Claim 25, wherein:

said step of providing a branch address to the program counter left shifts a signed displacement designated by the displacement field of the conditional branch-decrement instruction by a predetermined amount and adds the left shifted signed displacement to current contents of the program counter.

30. (New) A method of operating a digital system having a microprocessor with a conditional branch-decrement instruction, comprising the steps of:

fetching a conditional branch-decrement instruction for execution;

testing a test register selected by the conditional branch-decrement instruction to determine if the contents of the test register meet a first condition;

providing a branch address to a program counter to cause a branch by combining a displacement field of the conditional branch-decrement instruction with current contents of the program counter if the contents of the test register meet the first condition; and

modifying the contents of the test register if the contents of the test register meet the first condition.

31. (New) The method of Claim 30, further comprising the steps of:

storing data in a register file including a plurality of distinct general purpose registers;

recalling data from an instruction designated general purpose register for supplying an operand to a functional unit;

storing destination data generated by a functional unit in an instruction designated general purpose register; and

designating via the conditional branch-decrement instruction one of said general purpose registers as said selected test register.

32. (New) The method of Claim 30, wherein:

said step of providing a branch address to the program counter adds a signed displacement designated by the displacement field of the conditional branch-decrement instruction to current contents of the program counter.

33. (New) The method of Claim 30, wherein:

said step of providing a branch address to the program counter left shifts a signed displacement designated by the displacement field of the conditional branch-decrement instruction by a predetermined amount and adds the left shifted signed displacement to current contents of the program counter.